JAVA PRINTER

Publication number: WO9743720
Publication date: 1997-11-20

Inventor: LEE KANGHOON
Applicant: RICOH CORP (US)

Classification:
- international: ***B41J29/38; G06F3/12; G06K15/00;*** **B41J29/38; G06F3/12; G06K15/00;** (IPC1-7): G06F15/00
- european: G06F3/12T; G06K15/00
Application number: WO1997US07649 19970514
Priority number(s): US19960017398P 19960514

Also published as:

EP0979458 (A1)
JP2006202303 (A)
JP2004046908 (A)
EP0979458 (A4)
EP0979458 (A0)

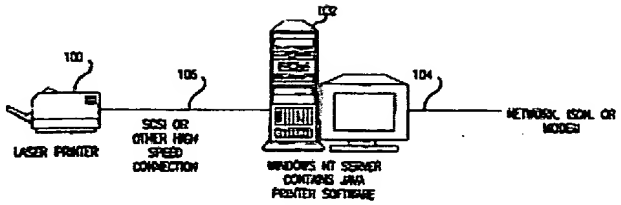
Cited documents:

US5469373
US5165014
US5371837
US5075874
US5228118
more >>

Report a data error here

Abstract of **WO9743720**

A method and system for printing documents based on Java commands. A Java printer (figure 1, item 100) receives page layout request (figure 3, item 131) and converts the request into a rasterized image which is transferred to a recording medium. Page layout can be interactively modified. The Java printer (figure 1, item 100) also monitors print request (figure 4, item 4) and is configurable using a World Wide Web interface (figure 3, item 128).



PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 15/00 | A1

(11) International Publication Number: WO 97/43720

(43) International Publication Date: 20 November 1997 (20.11.97)

(21) International Application Number: PCT/US97/07649

(22) International Filing Date: 14 May 1997 (14.05.97)

(30) Priority Data:
60/017,398 14 May 1996 (14.05.96) US

(71) Applicant: RICOH CORPORATION [US/US]; TQA Division, System Products Business, 3001 Orchard Parkway, San Jose, CA 95134 (US).

(72) Inventor: LEE, Kanghoon; 47112 Warm Springs Boulevard #215, Fremont, CA 94539 (US).

(74) Agents: SPIVAK, Marvin, J. et al.; Oblon, Spivak, McClelland, Maier & Neustadt, P.C., Crystal Square Five, 4th floor, 1755 Jefferson Davis Highway, Arlington, VA 22202 (US).

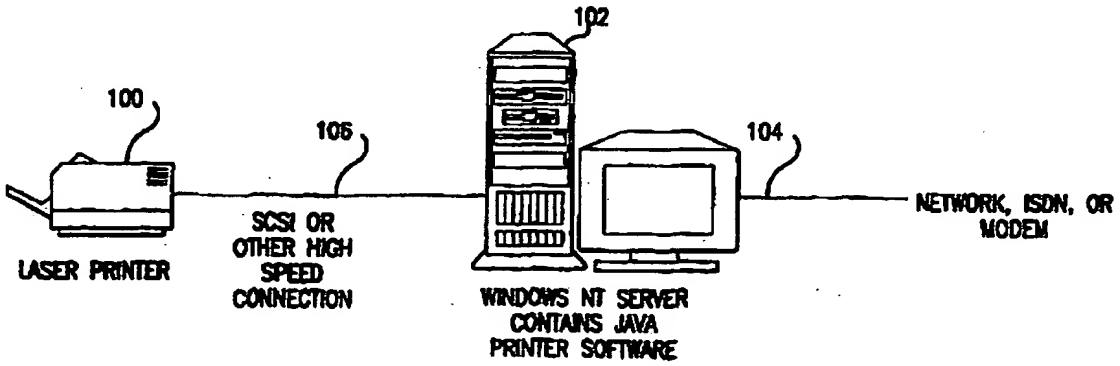
(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, UZ, VN, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published

With international search report.

Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: JAVA PRINTER



(57) Abstract

A method and system for printing documents based on Java commands. A Java printer (figure 1, item 100) receives page layout request (figure 3, item 131) and converts the request into a rasterized image which is transferred to a recording medium. Page layout can be interactively modified. The Java printer (figure 1, item 100) also monitors print request (figure 4, item 4) and is configurable using a World Wide Web interface (figure 3, item 128).

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

JAVA PRINTER

BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

This invention relates to a printer or printer system using the Java language to control rasterizing an image and to control printing.

DESCRIPTION OF THE BACKGROUND

The Internet is undergoing explosive growth and many new technologies are being developed to keep up with this growth. Previously, in order to develop applications quickly, application developers sought to use specialized cross-platform application development techniques which create applications for multiple hardware and software platforms. For example, operating systems (i.e. Solaris, Windows 95, Windows 3.x, OS/2 and Unix) create applications and executable files differently and have been implemented on various processor types (Intel processors, 680x0, Power PCs, and Sun SPARCs). To develop applications for all permutations of operating systems and hardware is difficult and extends the product development cycle. To make cross-platform development faster and easier, Sun Microsystems developed a language called Java which is object-oriented but simple. Java is described in Java in a Nutshell: A Desktop Quick Reference for Java Programmers by David Flannagan, published by O'Reilly & Associates, Inc., as well as in the Sun Series of books published by Prentice Hall Books entitled Core Java, Instant Java, Java by Example, and Just Java by Cornell, Pew, Jackson and Van Der Linden, respectively, which are incorporated herein by reference. One of Java's advantages is that it is a portable language which is independent of

it is a portable language which is independent of operating systems and hardware architectures. Further, applications developed using Java are adaptable or extendable using Java's ability to download new classes dynamically and to add the downloaded classes to an existing class hierarchy. Java also provides the advantages of distribution, language interpretation, security, high performance and a multi-threaded implementation.

Java enables applications to be written using an extensible set of objects, with each set of objects being defined in a separate group of objects called a package. The core set of objects for Java are defined in the java.lang package and describe the most central characteristics of the Java language. One of Java's advantages is that the character type that Java uses is the Unicode standard which allows English and Asian characters to be represented consistently and together in applications or documents generated using Java.

Other languages have been used to represent the layout of documents as they appear on printers. PostScript by Adobe is an extendable page-layout language which supports text and graphics on the same page. Some aspects of PostScript are described in PostScript by Example by Henry McGilton and Mary Campione, published by Addison-Wesley Publishing Company, the contents of which are incorporated herein by reference. PostScript uses stacks and dictionaries to extend the language. Some PostScript printers also have been equipped with non-volatile memories which are used to store configuration information for the printers. However, PostScript's lack of strong security features has enabled malicious users of the printer to update the parameters stored in the non-

volatile memory, thereby disturbing the printer's use in network environments.

Other printer languages, such as PCL by Hewlett-Packard, have evolved from uni-directional, dot-matrix line printers and therefore lack many of the operators needed to control the placement of images on a page. PCL also lacks modularity. The macros defined by PCL use globally scoped variables that can affect the performance of other macros defined by the language.

Currently, all of the applications that print from host systems have to convert their internal document format to PostScript or PCL and then download the document to the printer using a printer driver designed to work with the specifics of the connected printer. Since there is a wide variety of printers that can be used, each with a slightly different set of features and/or bugs, a large number of printer drivers have traditionally been shipped with applications, even though end-users actually only need the printer drivers for their specific printers. Furthermore, using conventional printing techniques, an inadvertent change in the printer driver used could cause the printer to print out the commands which describe how a page is to be laid out rather than interpreting the commands and rendering a resulting image. Further, for printers which support downloadable fonts, downloading of fonts often has been restricted to downloading to the printer's RAM, ROM font cartridge or attached hard disk.

This model of application and printer driver interaction has created an increase in work performed by end-users because of the inflexibility and limited communication capability of the printer when communicating with the application.

SUMMARY OF THE INVENTION

It is an object of the present invention to overcome at least one of the deficiencies described above in the implementation of a printer language.

It is a further object of the present invention to provide a printer which uses the Java language to interpret page-layout requests.

It is another object of the present invention to provide bi-directional communication between an application or printer driver and a Java printer to enable a user to define how a page or series of pages should be laid out.

It is yet another object of the present invention to provide a World Wide Web interface to control a Java printer of the present application.

It is a further object of the present invention to use the Java-specific features of object-orientation, distribution, interpretation, security, architecture and neutrality, portability, performance, multi-threadedness, and dynamic loading to implement an improved printer.

conventional laser printer 100 is connected to a print server 102 via a high speed communication link 106 (i.e. SCSI bus), and print server 102 receives jobs via an external communication link 104 which can be a network link (Ethernet, token ring, ATM), an ISDN connection, or a modem connection. The print server 102 can be any general purpose computer system capable of running a Java interpreter, and specifically includes at least a central processing unit (CPU), random access memory (RAM), a mass storage device (i.e., a hard disk, a magneto-optical disk), an input device (i.e., mouse, keyboard, touch screen) and an output device (i.e., monitor, heads-up display, virtual reality headset). Further, the print server 102 includes any of the available commercial operating systems (i.e., UNIX, Windows 95, Windows NT, OS/2, Linux) and any other software required to implement network or Internet communications.

In the first embodiment, Java print requests are received by the print server 102 via the external communication link 104 through a socket listening on a designated port (i.e., port 80 for hypertext transfer protocol), and the print server 102 converts the received print request from a Java request to a printer request for the attached laser printer 100 in the printer language of the attached laser printer 100. The print server 102 also can receive standard hypertext transfer protocol (HTTP) requests and produce World Wide Web (WWW) pages as a result or update the configuration of the laser printer 100 if the HTTP request has been sent by a system administrator or an authorized user.

As shown in Figure 2, the laser printer 100, print server 102 and communication link 106 can be combined into a single network printer which is a Java

printer 110 connected to the external communication link 104. In the second embodiment, the Java printer 110 receives print requests or WWW/HTTP requests directly by listening on the appropriate port. Since Java is architecture neutral, the creation of a Java printer 110 and its corresponding classes can quickly be ported or migrated to a new processor used in a new laser printer when the new processor becomes available. Further, when Java processors become available which execute Java Unicode instructions directly, the Java printer 110 can be implemented with a Java processor. Any Java printer 110 should conform to the Java Virtual Machine Specification. The August 21, 1995, Release 1.0 Beta Draft version of the JVM specification is incorporated herein by reference.

Java has a rich set of graphics operators that match PostScript and PCL, and since Java is an object-oriented language which provides extensions, new complex graphics operators can be created which are subclasses of the existing graphics primitives, thereby allowing complex images to be described compactly. The present invention extends the Graphics class of the java.awt package to control drawing of images on pages, by implementing a new class, Printer. Each of the other methods of the Graphics class would likewise be implemented to allow colors and fonts to be changed and to allow lines and filled and empty polygons to be drawn. For example, calling java.awt.Printer.drawString(msg, x, y) would cause the string "msg" to be drawn on a page at position (x,y). The Printer class also would implement a function, similar to PostScript's "showpage" command, which signals that a complete page has been rendered and that the resulting image should be transferred to Java laser printer 110. Further, based on the built-

in security, network capability and multi-lingual support of Java, a Java printer can handle inputs from different platforms in multiple languages and create the desired documents.

As shown in Figure 3, a system administrator who wishes to configure a Java printer 110 can use a standard WWW browser (i.e., Netscape Navigator, Mosaic, Microsoft Navigator, IBM Web Explorer) to remotely configure the Java printer 110. The Web browser screen 120 includes a title bar 122, a menu bar 124, button icons 126, a document identifier text entry box 128 and plural additional controls such as drop box 130 which includes a list of Internet connections for which a configuration can be established. A system administrator would authenticate himself/herself to the Java printer 110 using any available security technique (i.e., Secure Socket Layer, Public Key Encryption, Symmetric Key Encryption, or a User ID and Password Hash), wherein the public key, private key or valid user ID and password hashes are stored in the non-volatile memory of the Java printer 110 or in remote local with which the Java printer 110 can communicate securely. Having authenticated himself/herself, the system administrator would choose the configure document of printer1 as the document to be opened by specifying the appropriate URL in the text entry box 128, such as specifying:

http://printer1.companyname.com/configure.

When the Java printer 110 which was listening on the appropriate port receives the request for this document, the Java printer 110 would parse the name of the document requested from the rest of the request and determine that the system administrator wished to configure the printer for a particular user. (The

complete specification for the format of an HTTP request or response can be found in the HTTP standard (versions 1.1 or 1.0). Information on HTML can be found in Using HTML: The Definitive Guide by Musciano and Kennedy, and information on Internet information services can be found in Managing Internet Information Services by Liu et al. The contents of these references are incorporated herein by reference.) In response to a request for a document, the Java printer 110 would send back to the browser, via the socket used to send the request, a response like the page shown in Figure 3 and referenced by reference numeral 131. The Java printer 110 also may use other information in the request to determine if the default response should be modified. The response could be modified to include text in the system administrator's native language, or similar language/cultural changes. The system administrator could then choose which user's Internet number the configuration is for by selecting an Internet number from combination box 130. As would be evident to one of ordinary skill in the art, this combination box could be replaced by a select box such that multiple configurations for multiple Internet addresses could be set simultaneously. The returned page 131 allows the system administrator to establish paper size by choosing one of the radio buttons selecting A4 paper, 8.5 x 11 paper or legal size paper. Also, the default tray is specified, allowing the system administrator to choose between letterhead paper and plain paper.

Additional values can be set, as shown in the Figure 3 and as well as those parameters that would be evident to one of ordinary skill in the art in light of the present invention. In fact, any parameter usually set by buttons on a printer can be set through

the standard WWW interface. In addition, on-line information which can not be easily displayed using LCD screens on the printer can be provided by selecting a hypertext link 134. This help information can either point to local help or remotely stored help, such as might be stored at the WWW site of the manufacturer of the Java printer 110.

Further, the Java printer 110, or the combination print server 102 and laser printer 100, can utilize the external communication link 104 dynamically to retrieve documents or portions of documents from other web sites for printing or for otherwise modifying the operation of the Java printer 110, such as loading updates to printer code. The Java printer 110 can likewise use the external communication link 104 for downloading fonts from remote locations whenever the font is needed. To download fonts, changes, or other information, the Java printer 110 can use any transfer protocol implemented by an included Java package, including HTTP, FTP, Gopher, etc. By specifying the URL from which the information should be obtained and using the java.net.getContent() method information can be received as a String. In addition, for new or previously unimplemented protocols, a java.net.URLConnection object would be used to receive and parse the contents of a new document type. As described earlier, this provides an advantage over PostScript printers which have to have fonts loaded in their RAM, ROM or hard disk in order to use them. Table 1 below shows some of the advantages of using Java over HP/PCL and PostScript.

Table I

Network Configuration	Partially	Partially	Yes
Network Security	No	No	Yes
Code Size	Small	Big	Small
Ability to Handle Double-Byte Characters	Difficult	Difficult	Easy
Customization	Difficult	Difficult	Easy
Resources	One Location	One Location	Anywhere on the Network
Printer Driver	Printer Specific	Printer Specific	Universal
Graphics Operation	Difficult	Medium	Easy
Extensibility	Difficult	Difficult	Easy
Upgrade	Difficult	Difficult	Easy
Remote Diagnostics and Maintenance	Very Difficult	Very Difficult	Easy

As shown in Figure 4, the Java printer 110 can additionally be configured with a queue manager which can further control the Java printer 110 using a standard WWW interface 120. Figure 4 shows that a system administrator has requested queue information from Java printer 110 and received back an applet which provides configuration information to the system administrator. When the system administrator uses URL:

http://printer1.companyname.com/queue,

the web browser 120 changes the title bar 122 to reflect that the requested page represents the queue manager. The returned page 136 includes a series of graphics 160 which represent documents, their titles,

their print times and the Internet addresses from which the jobs were submitted. These graphics 160 are displayed in a scrollable window 144 controlled by a scroll bar 138. Further, the applet includes a filter with an associated combination box 140 for filtering the graphics 160 which are displayed in the scrollable window 144. As shown in Figure 5, when an Internet address is used as a filter, all documents not submitted by a specified Internet address (i.e., 123.45.67.89) are removed from the scrollable window 144. Referring back to Figure 4, the applet also provides buttons 146, 148, 150, 152, and 156 which control how the Java printer 110 processes the documents. To control a print job, any of the graphics 160 can be highlighted and then one of the buttons 146, 148, 150, 152 and 156 depressed. The queue manager then will modify the queue characteristics for the highlighted job(s) according to the button which was depressed.

Since Java is multi-threaded, the Java printer 110 can multi-task between any of its functions (i.e., printing the current job, receiving a new job, pausing a job, killing a job, reordering the job such that a job is made the next job to be printed or the last job to be printed, displaying characteristics of a job, resuming a paused job, setting default configuration information or determining the status of the printer). Conventional printers perform one job at a time and do not have such a printing capability.

As shown in Figures 4 and 5, another advantage of a Java printer 110 is that status information 139 can be requested while other activities are being performed. Although the status information 139 is only a snapshot of the current status, the applet can periodically update the status information 139 using

the network capabilities of Java and the browser provided by the DatagramPacket, DatagramSocket and Socket classes, as implemented by the java.net package. Furthermore, since the Java printer 110 performs active multi-tasking, the Java printer 110 can also periodically send status reports via to system administrators or other users by any other implemented protocol (i.e., SMTP, FTP).

As shown in Figure 6, the Java printer 110 is also capable of performing interactive pagination based on an application using Java as the print language. The Java printer 110 can either be used as a traditional printer whereby the application specifies the pagination and sends print requests to the Java printer 110 in preformatted pages, or the application wishing to print uses the enhanced capabilities of the Java printer 110 to request that the Java printer 110 create a default pagination based on the characteristics of the printer.

Figure 6 shows a document which has been paginated by a Java printer 110. Having sent Java code in the form of a WWW/HTTP request to the Java printer 110, the pagination applet, or a portion of the application written to support Java, displays the result as specified by the Java printer 110. In Figure 6, the document has been split into four thumbnail sketches 170 which depict the layout of the document as established by the Java printer 110. The four thumbnail sketches 170 show that a figure is split across the second and third thumbnail sketches 170 into two parts 172a and 172b. Further, a paragraph of text has been split between the first and second pages, leaving a single line 171a separate from the rest of the text 171b. To modify the default pagination, the application places page arrows 173

next to the thumbnail sketches 170 to indicate where the user wishes the pagination to actually occur. Using page arrows 173, the single line of text of 171a can be grouped with the rest of the text 171b, and the image 172a and 172b can be grouped. When the user has set the page arrows 173, the user uses the Resubmit button 174 to send the new configuration Java printer 110 which recreates thumbnail sketches 170 and returns the new result to web browser or pagination output/application. If the thumbnail sketches 170 were satisfiable to the user, the Ok button 176 would be used to signal that the pagination has been accepted by the user and that the pages should be printed. In addition, the Cancel button 178 can be depressed if the user wishes to cancel the printing. This allows the Java printer 110 to clear the Java print request from the pending queue of outstanding paginations. By explicitly clearing this information, the garbage collection process of the Java interpreter can reclaim the memory of the Java printer 110 more quickly. Although Figure 6 has been described in terms of page arrows 173, any type of pagination identifier could be used to describe how pages should be repaginated. For example, a continuous, scrollable image could be presented, and the user would use the applet/application to draw lines where the pagination should be. Furthermore, the standard hard return tag of HTML, <HR>, can be used in an alternate embodiment to show where page breaks occur. In addition, the thumbnail sketches 170 can be any type of graphic that can be displayed on a page 120. These graphics include thumbnail GIFs, Java images, etc., several of which are supported by the java.image package.

As shown in Figure 7, the applet of Figures 4 and 5 can be generated using standard HTML code with a

browser which supports Java. Using the APPLET keyword, the browser knows to load the queue manager class and display the window with a width of 500 and a height of 300.

As shown in Figure 8, the Java printer 110 can be extended with new classes such as the Letterhead class of Figure 8. The Letterhead class extends the printing class which provides the base printing functionality of the present invention. Although the Letterhead class shown is not completely illustrated, it would be evident to one of ordinary skill in the art that the Letterhead class would be used to first print the letterhead text onto a page to be printed, and then the Print class would perform its printing of the rest of the page. The illustrative class allows the printing on two types of letterhead, cover sheets and everything else, by specifying different strings and positions for the two pages. The Letterhead class could be extended in an alternate embodiment to use arrays, as supported by the Java language, to allow multiple strings and positions to be stored for each letterhead page.

Furthermore, to smooth the transition from PostScript and PCL to Java, Java printers can be implemented which utilize subclasses of the Printer class that receive and render PostScript, PCL or both. These classes can even be added dynamically by the system administrator when they become available. The new classes would be loaded using a URL specified by the system administrator. On the otherhand, traditional printer drivers which utilize GDI calls can be modified to generate Java code from the received GDI application. For example, since windows documents are generated using standard GDI calls, a Java printer 110 can be selected like any other

supported printer, and the Java printer driver then would convert the Windows GDI calls into Java code to be sent to the Java printer 110.

Obviously, numerous modifications and variations of the present invention are possible in light of the above teachings without departing from the intended scope of the present invention. Such changes include storing the user profile information in remote databases rather than in the Java printer 110 itself. By using the Java SQL API, also known as the JDBC, remote databases can store the user profile information, and applets still can query and update the user profile information. Since the JDBC specification has not been finalized, the version 0.70 draft specification dated May 7, 1996 is incorporated herein by reference.

Claims:

1. A computer program product, comprising:

a computer storage medium and a computer program code mechanism embedded in the computer storage medium for causing a printer to control rasterization of an image, the computer program code mechanism comprising:

a first computer code device configured to receive a print request as a series of Java commands;

a second computer code device configured to rasterize the series of Java commands into an image; and

a third computer code device configured to output the image on a recording medium.

2. The computer program product as claimed in claim 1, wherein the third computer code device comprises a fourth computer code device configured to receive an end of page command and to output the image after receiving the end of page command.

3. A computer program product, comprising:

a computer storage medium and a computer program code mechanism embedded in the computer storage medium for causing a printer to control a configuration of the printer, the computer program code mechanism comprising:

a first computer code device configured to receive a request for a printer control interface;

a second computer code device configured to send the printer control interface to a remote computer;

a third computer code device configured to receive a series of printer control parameters in response to sending the printer control interface; and

a fourth computer code device configured to update a control memory of the printer based on the series of printer control parameters.

4. The computer program product as claimed in claim 3, wherein the fourth computer code device comprises a fifth computer code device configured to store the series of printer control parameters in the control memory of the printer based on an identification of the remote computer.

5. The computer program product as claimed in claim 3, wherein the fourth computer code device comprises a fifth computer code device configured to store the series of printer control parameters in the control memory of the printer based on an Internet address of the remote computer.

6. A computer program product, comprising:

a computer storage medium and a computer program code mechanism embedded in the computer storage medium for causing a printer to control a layout of a document to be printed by the printer, the computer program code mechanism comprising:

a first computer code device configured to generate an initial layout of a document in the printer;

a second computer code device configured to send the initial layout to a remote computer;

a third computer code device configured to receive pagination indications indicating where the initial layout is to be split into pages; and

a fourth computer code device configured to print out the pages as indicated by the pagination indications for the initial layout.

7. A computer program, comprising:

a computer storage medium and a computer program code mechanism embedded in the computer storage medium for causing a printer to control an order of printing, the computer program code product mechanism comprising:

a first computer code device configured to track pending print requests;

a second computer code device configured to send to a first remote computer the pending print requests;

a third computer code device configured to receive commands from the first remote computer; and

a fourth computer code device configured to control an order of printing based on the commands received by the third computer code device.

8. The computer program product as claimed in claim 7, wherein the first computer code device comprises a fifth computer code device configured to track pending print requests by an identification of which remote computer submitted substantially each of the pending print requests.

9. The computer program product as claimed in claim 8, wherein the second computer code device comprises a sixth computer code device configured to send only the pending print requests submitted by a second remote computer with a specified identification.

10. The computer program product as claimed in claim 7, wherein the third computer code device comprises a fifth computer code device configured to receive a command of at least one of: 1) killing a pending print request, 2) pausing a pending print request, 3) resuming a pending print request, and 4) changing a priority of a pending print request.

11. The computer program product as claimed in claim 7, further comprising a fifth computer code device operating in conjunction with the second computer code device and configured to send a status of the printer to the first remote computer.

12. The computer program product as claimed in claim 12,

wherein the fifth computer code device comprises a sixth computer code device configured to send at least one of: 1) a toner status, 2) a paper jam status, and 3) a number of printed pages.

13. The computer program product as claimed in claim 7, further comprising a fifth computer code device operating in conjunction with the second computer code device and configured to provide interactive help to a user of the first remote computer.

14. The computer program product as claimed in claim 7, wherein the third computer code device comprises a Java applet.

15. The computer program product as claimed in claim 7, further comprising a fifth computer code device configured to print pending print requests,

wherein the third, fourth and fifth computer code devices are executed concurrently using multi-tasking.

16. The computer program product as claimed in claim 7, further comprising a fifth computer code device configured to print pending print requests,

wherein the third, fourth and fifth computer code devices are executed concurrently using multi-tasking of respective threads on a uniprocessor.

17. The computer program product as claimed in claim 7, further comprising a fifth computer code device configured to receive code updates from a second remote computer.

18. The computer program product as claimed in claim 7, further comprising a fifth computer code device configured to download a font from a second remote computer.

19. The computer program product as claimed in claim 7, further comprising a fifth computer code

device configured to periodically and autonomously send status reports to a system administrator.

20. The computer program product as claimed in claim 3, wherein the fourth computer code device comprises a fifth computer code device configured to store the series of printer parameters in a database.

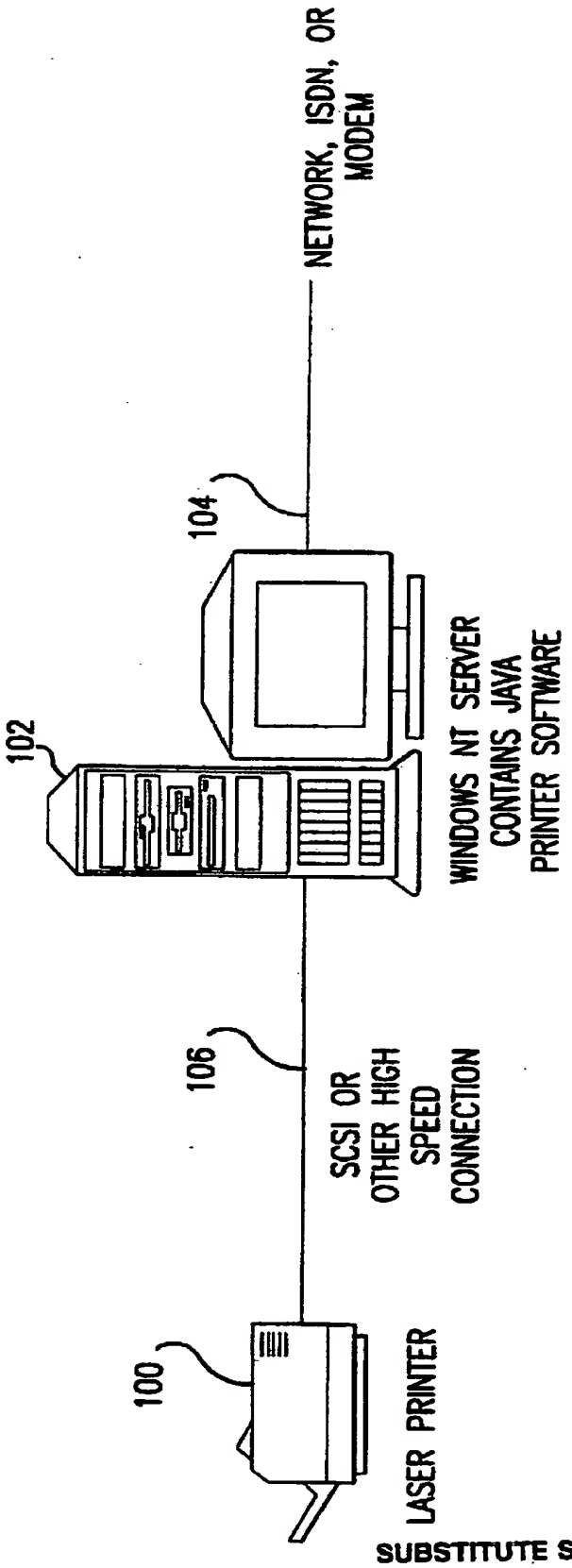

NETWORK, ISDN, OR MODEM
104
102
WINDOWS NT SERVER CONTAINS JAVA PRINTER SOFTWARE
106
SCSI OR OTHER HIGH SPEED CONNECTION
100
LASER PRINTER

FIG.1

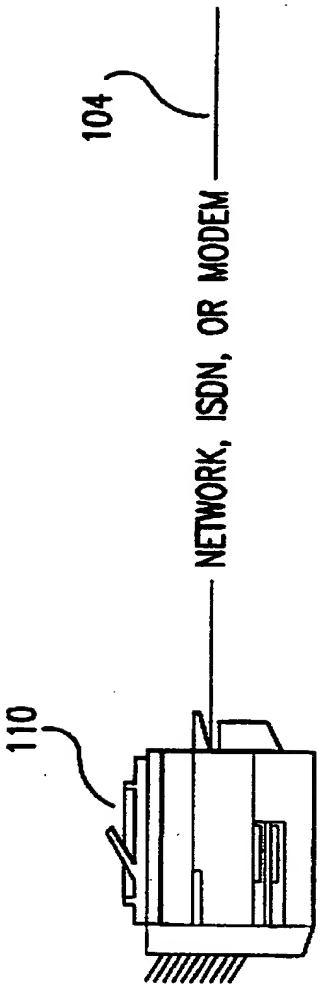

104
NETWORK, ISDN, OR MODEM
110

FIG.2

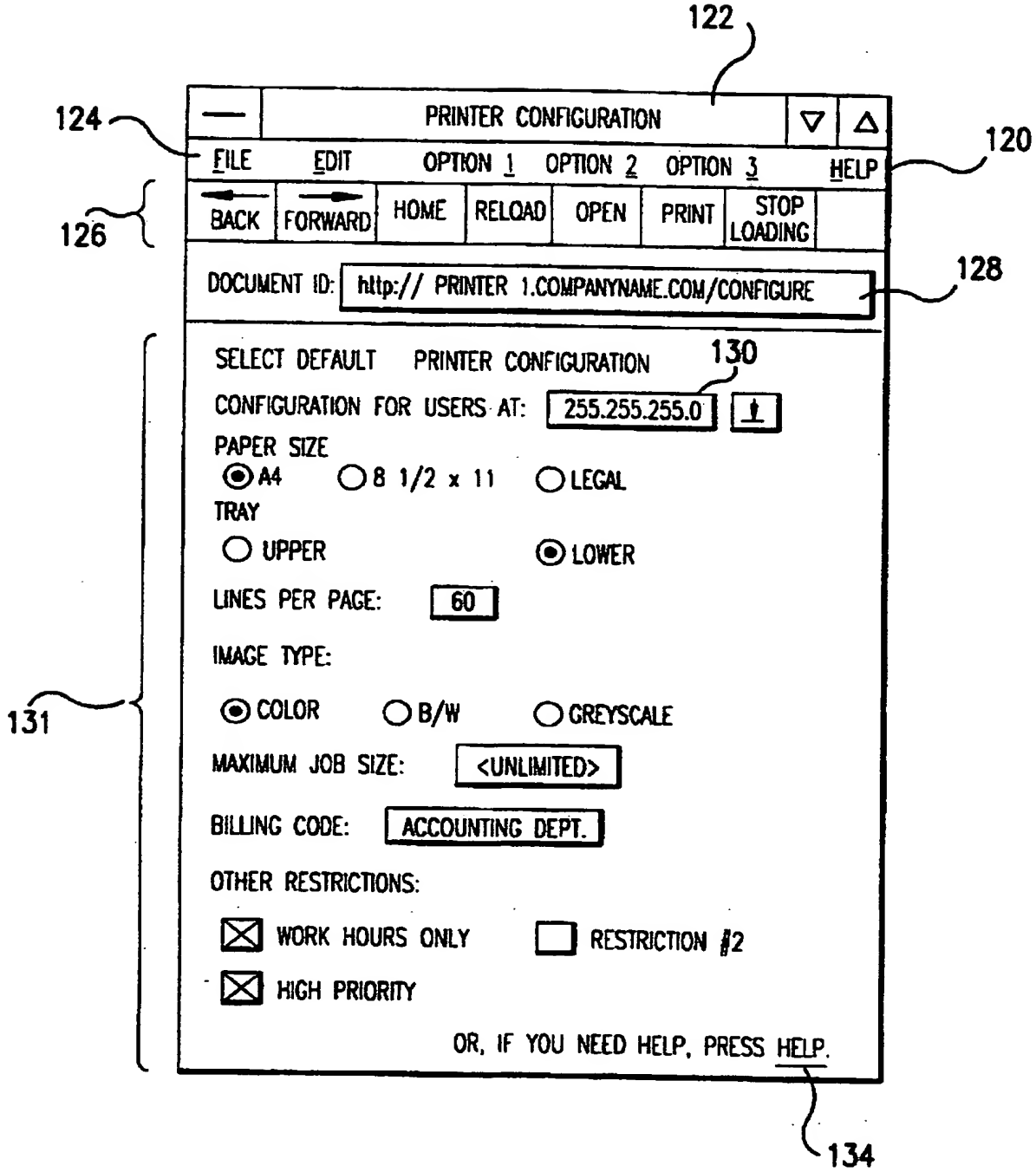

122
124
PRINTER CONFIGURATION
FILE EDIT OPTION 1 OPTION 2 OPTION 3 HELP
120
126
BACK FORWARD HOME RELOAD OPEN PRINT STOP LOADING
DOCUMENT ID: http:// PRINTER 1.COMPANYNAME.COM/CONFIGURE
128
SELECT DEFAULT PRINTER CONFIGURATION
130
CONFIGURATION FOR USERS AT: 255.255.255.0
PAPER SIZE
A4 8 1/2 x 11 LEGAL
TRAY
UPPER LOWER
LINES PER PAGE: 60
IMAGE TYPE:
131
COLOR B/W GREYSCALE
MAXIMUM JOB SIZE: <UNLIMITED>
BILLING CODE: ACCOUNTING DEPT.
OTHER RESTRICTIONS:
WORK HOURS ONLY RESTRICTION #2
HIGH PRIORITY
OR, IF YOU NEED HELP, PRESS HELP.
134

FIG.3

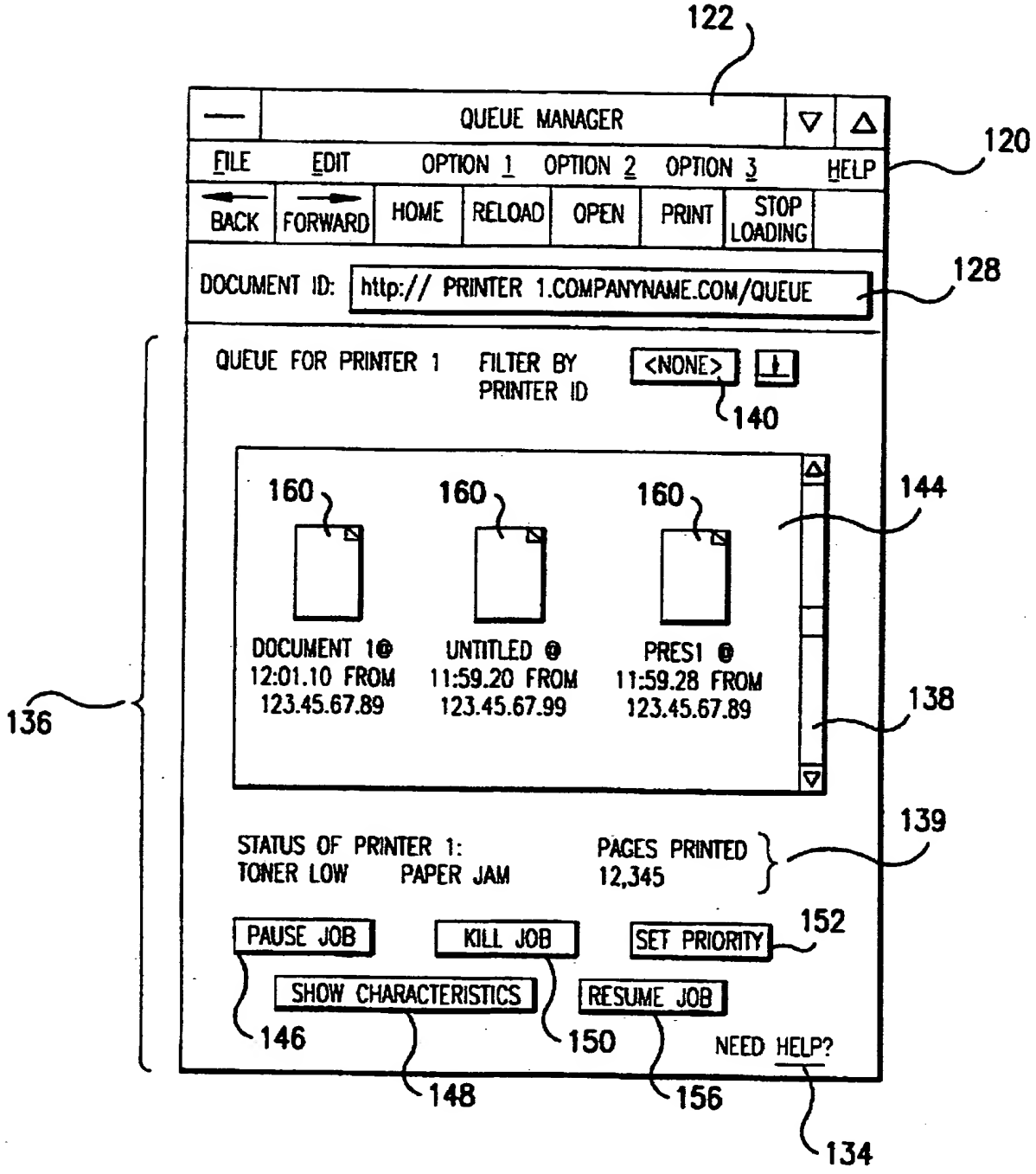

122
QUEUE MANAGER
120
FILE EDIT OPTION 1 OPTION 2 OPTION 3 HELP
BACK FORWARD HOME RELOAD OPEN PRINT STOP LOADING
DOCUMENT ID: http:// PRINTER 1.COMPANYNAME.COM/QUEUE
128
QUEUE FOR PRINTER 1
FILTER BY PRINTER ID
<NONE>
140
160
160
160
144
DOCUMENT 1@ 12:01.10 FROM 123.45.67.89
UNTITLED @ 11:59.20 FROM 123.45.67.99
PRES1 @ 11:59.28 FROM 123.45.67.89
136
138
139
STATUS OF PRINTER 1:
TONER LOW PAPER JAM
PAGES PRINTED 12,345
PAUSE JOB
KILL JOB
SET PRIORITY
152
SHOW CHARACTERISTICS
RESUME JOB
146
150
148
156
NEED HELP?
134

FIG.4

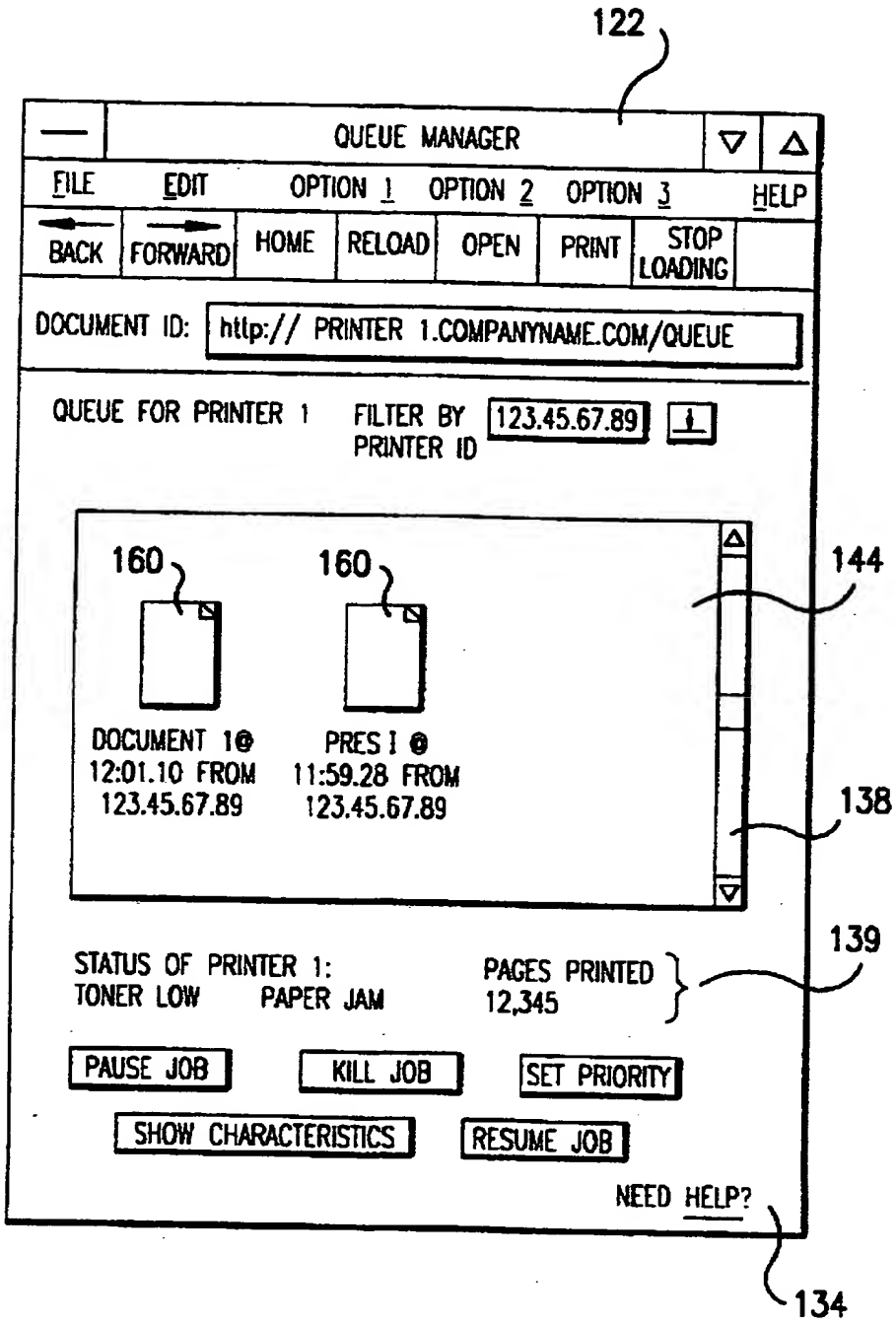

122
QUEUE MANAGER
FILE
EDIT
OPTION 1
OPTION 2
OPTION 3
HELP
BACK
FORWARD
HOME
RELOAD
OPEN
PRINT
STOP LOADING
DOCUMENT ID:
http:// PRINTER 1.COMPANYNAME.COM/QUEUE
QUEUE FOR PRINTER 1
FILTER BY PRINTER ID
123.45.67.89
160
160
144
DOCUMENT 1@ 12:01.10 FROM 123.45.67.89
PRES I @ 11:59.28 FROM 123.45.67.89
138
139
STATUS OF PRINTER 1:
TONER LOW
PAPER JAM
PAGES PRINTED 12,345
PAUSE JOB
KILL JOB
SET PRIORITY
SHOW CHARACTERISTICS
RESUME JOB
NEED HELP?
134

FIG.5

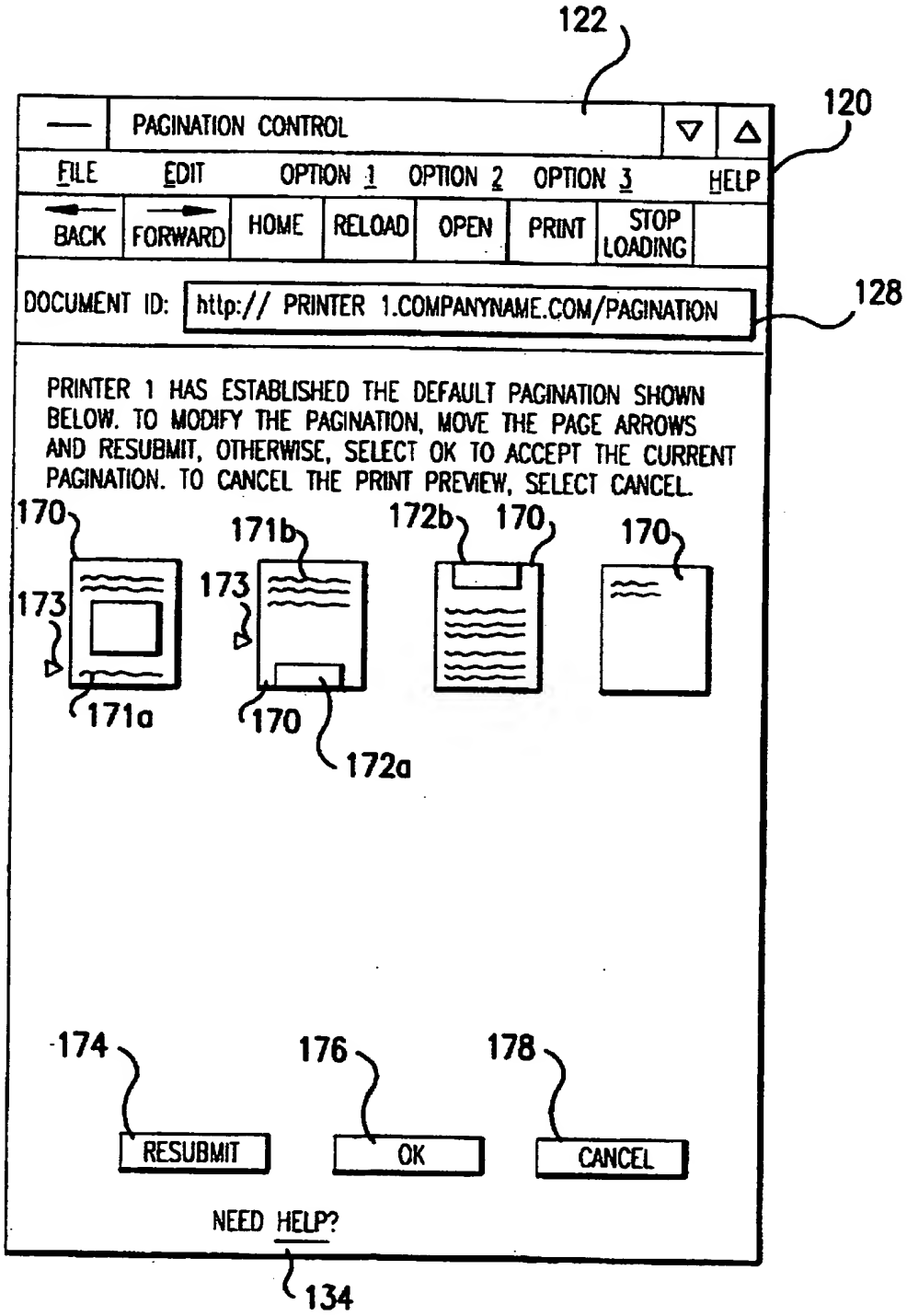

122
120
PAGINATION CONTROL
FILE EDIT OPTION 1 OPTION 2 OPTION 3 HELP
BACK FORWARD HOME RELOAD OPEN PRINT STOP LOADING
DOCUMENT ID: http:// PRINTER 1.COMPANYNAME.COM/PAGINATION
128
PRINTER 1 HAS ESTABLISHED THE DEFAULT PAGINATION SHOWN BELOW. TO MODIFY THE PAGINATION, MOVE THE PAGE ARROWS AND RESUBMIT, OTHERWISE, SELECT OK TO ACCEPT THE CURRENT PAGINATION. TO CANCEL THE PRINT PREVIEW, SELECT CANCEL.
170
171b
172b
170
170
173
173
171a
170
172a
174
176
178
RESUBMIT
OK
CANCEL
NEED HELP?
134

FIG.6

```
<HTML>
<HEAD>
<TITLE> QUEUE MANAGER <TITLE>
</HEAD>
<BODY>
<APPLET CODE = "QUEUEMGR.CLASS" WIDTH = 500 HEIGHT = 300>
</APPLET>
</BODY>
</HTML>
```

FIG.7

```
PUBLIC CLASS LETTERHEAD EXTENDS PRINTER {

 PROTECTED STRING  PAGE_1_TEXT;
 PROTECTED STRING  PAGE_2_TEXT;
 PROTECTED int  p1x,p1y,p2x,p2y;
 PUBLIC  SET PAGE1 TEXT (STRING s1)  {PAGE_1_TEXT = s1;}
 PUBLIC  SET PAGE2 TEXT (STRING s2)  {PAGE_2_TEXT = s2:}
 PUBLIC DRAWPAGE (int PAGE_NUMBER)
   IF (PAGE_NUMBER = = 1)
    {DRAW STRING (p1x, p1y, PAGE_1_TEXT); SUPER, DRAWPAGE (PAGE_NUMBER):}
   ELSE
    {DRAW STRING (p2x, p2y, PAGE_2_TEXT); SUPER, DRAWPAGE (PAGE_NUMBER):}
   }
 :
 // OTHER DEFINITIONS OF METHODS OR DATA MEMBERS
}
```

FIG.8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US97/07649

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :G06F 15/00
US CL : 395/114

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 395/101, 109, 112, 114, 117, 135, 169, 333, 342, 433, 570

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

DIALOG files 9, 12, 15, 16, 47, 75, 88, 148, 237, 256, 275, 278, 621, 624, 636, 647, 674

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,469,373 A (KASHIWAZAKI et al.) 21 November 1995, the whole document	1-20
Y	US 5,165,014 A (VASSAR) 17 November 1992, the whole document	1-20
Y	US 5,371,837 A (KIMBER et al.) 06 December 1994, the whole document	1-20
Y	US 5,075,874 A (STEEVES et al.) 24 December 1991, the whole document	1-20
Y	US 5,228,118 A (SASAKI) 13 July 1993, the whole document	1-20
Y	US 5,293,466 A (BRINGMANN) 08 March 1994, the whole document	1-20

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be part of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
14 AUGUST 1997	12 SEP 1997
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer GABRIEL I. GARCIA Telephone No. (703) 305-8751

Form PCT/ISA/210 (second sheet)(July 1992)*

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US97/07649

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y,E	US 5,638,497 A (KIMBER et al.) 10 June 1997, the whole document	1-20
Y,E	US 5,537,626 A (KRASLAVSKY et al.) 16 July 1996, the whole document	1-20
Y	US 5,121,113 A (KEDGE et al.) 09 June 1992, the whole document	1-20
Y	US 5,488,223 A (AUSTIN et al.) 30 January 1996, see whole document.	1-20
Y	US 4,642,792 A (CLEMENTS et al.) 10 February 1987, the whole document	1-20
Y	Jandel Scientific Announces Java, News Realease: Corte Madera, CA, 1 March 1991	1-20
Y,P	Bristol Technology Unveils First Cross Platform Java Printing Solution, Business Wire, 02 December 1996.	1-20
Y,P	Repeat/IBM Introduces Industry's First Java Application For Intranet Printer Management, Business Wire, 10 February 1997.	1-20
Y	COX, John, Printer Presses Forward with Java Application, Network World V13 n13 p41. 25 March 1996.	1-20
Y,P	IBM Re-Writes Printer Management Software in Java, Computergram International, 11 February 1997.	1-20